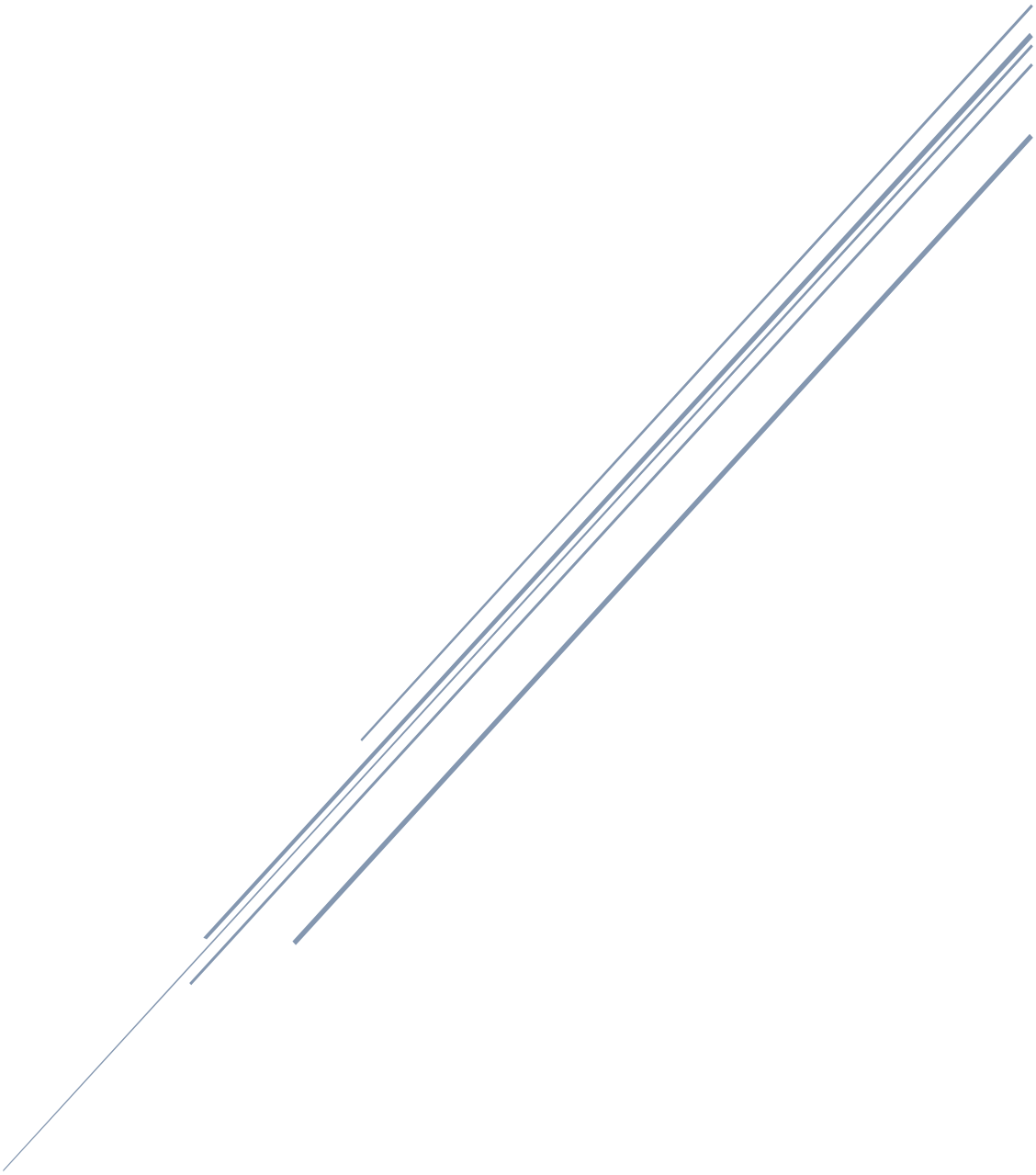
PERSONAL DEVELOPMENT REPORT

By: Arjen van der Meer



Fontys
Semester 6

Inhoud

Introduction 2
Learning goals 3
Developing and Deployment of Enterprise Software: 3
Context Based Research: 4
Preparation for Life-Long Learning: 5
Scalable Architectures: 7
Development and Operations (DevOps): 8
Cloud Services: 9
Security by Design: 10
Distributed Data: 11
Final conclusion ..... **Fout! Bladwijzer niet gedefinieerd.**
Semester reflection 12

Introduction

In this document I will be providing proof about the learning goals that are to be acquired in this semester. For each learning goal I will give a short introduction about what the learning goal means and then provide my proofs in table. This table will have a proof, location and reflection field, which means I will first tell you what my proof is, then where you can find it and lastly a reflection on what I learned from this proof. Finally I will write a conclusion about if I think I have done enough to complete the learning goal.

Note: Yellow marked bullet points are new since last PDR

Learning goals

Developing and Deployment of Enterprise Software:

Course definition:
You develop and deploy enterprise software both individually and as a team using a suitable enterprise development platform and application framework, following a professional software development process. You develop such software as a team effort taking into account both functional and non-functional requirements as set by the stakeholders as well as legislation.

- In the hackathon and in both sprints we have shown that we can work well in a team.
- We have a Youtrack server on which we track our progress by cards and by a burndown chart. We try and keep this as up to date as possible.
- Every week we do a standup on Monday to decide what we are going to work on in the coming week. We also check in on our progress daily to see what everyone is doing or what they could be doing.
- In one of our first meetings with the product owners we made a lot of user stories and non-functional requirements for our application.
- We have done an event storming session with Michiel. In this we learned a couple of small new things about our project and we were able to slightly expand and fine tune the user stories we already had.
- We have done multiple presentations of the work we have done in a sprint. And have discussed these results with our product owners and documented their feedback.
- We have made multiple retrospectives in which we discussed with each other what we thought we did right and what we thought we could have done better.
- I have made a page on the YouTrack that explains how to use the OpenDroneMapping Node service that we are currently running in our project.
- We have done multiple planning poker sessions to try and divide the workload as equally as possible among our team.
- I have reviewed merge request to ensure code quality is high and readable and start issues where I felt appropriate.
- Carefully read merge issues that were left by other reviewers and SonarQube and resolved these issues both in my own code and that of someone else.
- In a case study we have looked at the GDPR rules and how it could apply to our project. We have then written in what should be done to comply to these rules.

I think that I and the rest of the group have shown that we can properly work in group. I think that we work efficiently and try and keep the promises we made with each other as good as possible. That's why I think that this learning outcome is at proficient right now. But is going towards advanced as we work very well within our team and follow a lot of industry standards.

Context Based Research:

Course definition:
You substantiate your choices regarding processes and techniques using a commonly accepted research method and taking into account your own ethical values.

- We have done multiple case studies in which we have done extensive research using the DOT-framework. We then tried to draw a conclusion on what would work best for the given case study's scenario and what would work best in our own project.
- I have read all our written case studies that I didn't directly write to learn more of the different subjects that are important in this semester.
- I have done extensive research on deciding which ingress controller we were going to use for our group project. I have once again used the DOT-framework to research the different ingress controllers and made an extensive report on my findings and what would work best for our scenario.
- I have received feedback on my research of the ingress controllers and have applied this to the research document accordingly. I have also kept general feedback in mind when I was working on the research document for OpenDroneMapping.
- I have done extensive research on how to use OpenDroneMapping and how we could use it in our project. I then put this research into practice and learned how it can be applied within our project. Next I tried changing certain settings to see their effect on the runtime and the output OpenDroneMapping delivers. Finally I made a page on YouTrack where I explain how to use the OpenDroneMapping service in case other team members are going to be working with it.
- I have also started triangulating in my research as I have applied the things I found in my OpenDroneMapping research and documented my findings when I started working with it. As such I found that it would be best to build a adapter around the service we are using, to take away some of the complexity the service has, but also keep the service we are currently using as it does offer a lot of customization.

I think I am on proficient as I by now have done a lot of research for the different case studies and have written a lot of documentation for these case studies. On top of that I have done extensive research on OpenDroneMapping in our project and made a well explained research document for this.

Preparation for Life-Long Learning:

Course definition:
You signal emerging trends in software engineering, explore them, and apply them in your projects where appropriate. Also, you are aware of possible career paths and you acquire the required skills to be prepared for your future career.

- We have formulated multiple research questions that will help us in finding the best tools and implementations for our project. While doing so we can see if there are certain trends that we should try exploring to not keep on using the same tools and frameworks that we are used too.
- We have started working with Vue.js, a framework that none of us had experience with before. But we have decided to use this as we thought it looked very promising.
- I have worked with OpenDroneMapping which is an AI based program. As AI is a huge trend in programming it was a very interesting area to look at and see what kind of functionality it offers and how it performs. I have done this by myself and am currently the only member in my group project that has worked with OpenDroneMapping, however I have made a YouTrack page describing how they could use it if need be.
- I have added a text editor within our frontend that lets users craft good looking and clear notes. For doing so I used TipTap, which is an open source library that offers you a lot of customization when it comes to possible features. It is framework agnostic, meaning it would work in any framework as well as plain JavaScript. I had to make sure it was setup in such a way a user would not be able to abuse the editor by uploading and reading the content as html allowing for XSS.
- I have worked in sprint 3 mainly on frontend development but have also done some backend work as I have worked on fixing and extending an existing service that my frontend was using. Meaning I had to work on someone else's code. This will not be uncommon later in life and is good practice for learning to quickly read and understand someone else's code.
- I have started programming react with functional components as this is a trend within react as multiple developers are pushing for more use of function components instead of class components as this in general reduce the amount of code for the same functionality.
- I have gone further in working with node ODM as in started to work on a adapter to make the access to ODM easier and better integrated in the project.
- Started on working with python for decision theory. In this we are programming agents that can solve a game of minesweeper.
- For my graduation internship I would like to work with something I haven't worked with before. What this is I am not sure yet, but I want it to be in uncharted territory so I can grow my repertoire of different experiences within software development. I also still have an interest for AI that I would like to explore, preferably into the neural network side of it.
- In the future I see myself working as a full stack developer. Getting to know a lot of the different software, tooling and architectures that are being used to create business applications. My goal is, once I have enough knowledge, to try and become a lead architect. For this I need to be able to take the responsibility of managing a large scale project and be able to convince the people I'm working with that the project is being developed in the best way possible. To further accomplish this goal I want to achieve the follow-up master of the premaster that is currently included in my bachelor program.

I think this learning outcome is on advanced. As I took the objective of learning what OpenDroneMapping is and how we could apply it to our group project completely upon myself. On top of this I have worked with a lot of different frameworks and library's that I haven't worked with before. I tried to not stick to what I'm used to and try out new ways of programming the same things. I also have had a lot of experience with AI this semester that I haven't had before which has grown my current interest in AI. I think all of this is a good basis for my current career goal as I will need a lot of experience with different implementations of software.

Scalable Architectures:

Course definition:
You develop enterprise software based on a chosen distributed architecture that clearly supports scalability for high volume communication and event handling, and enables independent life cycle management.

- We have made design decisions in our architecture document to make sure our project can handle a high work load and consists of different services that can be run independent of each other.
- I have made a design for my individual project in such a way that I have separated functionality into different services that can be run independent of each other.
- We have a working implementation of Kubernetes running in docker where we have split up our projects functionality into different independent micro services.
- Added kafka communication for the project service within our group project so that it also can send the flower set that is saved to a specific project.
- I have my own working implementation of Kubernetes running in docker. It has multiple services that can be run at the same time and can communicate with each other using RabbitMQ when necessary.
- My auth service talks to my user service with a request reply pattern trough RabbitMQ. In this the auth service validates and creates JWT tokens, while the user service holds the information for users.
- I have implemented the ability for users to remove their account. By sending this request to the user service, the user service will send a message to the post service to remove all of the users data. This follows the event-driven pattern.
- Each time there is traffic on a post, a message is send to the trending service. The trending service keeps this traffic in memory till a certain threshold or time is reached and then writes them to the database. This ensures that data isn't constantly being written to the database but instead in is written in batches. These message also follow the event-driven pattern and no response is send back to the post service.
- The trending service keeps track of the trending posts during certain time frames. Due to this not changing very often and getting the trending posts is the same for each user this result will be cached in memory with Redis. These caches will be cleared based on what time frame they get. In practice this means that the daily trending posts will be cached for only 5 minutes. While the yearly trending posts will only be cleared every week. This minimizes the workload of this service, while keeping its functionality.
- Written documentation for the messages being send trough RabbitMQ. This documentation tells you what services are sending and receiving the message, what it is doing and what data is being send.
- I have made integration tests using postman to test if the different services correctly talk to each other and are doing what they are supposed to be doing. I have done this both for the user and auth communications as well as the posts and user communications.

I think this learning outcome is on proficient as I have my own Kubernetes implementation with multiple services talking to each other with RabbitMQ. For these communications I have implemented the request reply pattern as well as the event driven pattern and have written integrations tests using postman to test these communications. I also have written documentation for my services so it is more clear how my different services work, how they are connected to each other and how they communicate with each other.

Development and Operations (DevOps):

Course definition:
You set up an environment and team processes supporting a fully automated software life cycle, while ensuring high quality, high availability, fast delivery, and short release times.

- We have a working pipeline, where when we want to merge with the development branch, the code will be automatically build, tested and quality and style checked before it may actually be merged with the branch.
- I have set up my own Github runner so that I am able to build, test, scan and deploy my project when I have made a new push to the main development branch.
- I have SonarQube running on my own server to scan my project for possible code smells and security issues. It generates a report each time the runner asks it too and doesn't fail a previous task. Which is whenever I want to merge a branch into the main branch or push something in the main branch.
- I have created my own Github actions workflow that builds, tests and scans my project using SonarQube. I am not allowed to merge a request if it doesn't pass all the stages, ensuring my project is functional and has a certain amount of code quality.
- I have made sequence diagrams that show how services communicate and what actions they perform when a user wants to perform certain actions.
- I have set up the metrics service within my Kubernetes docker implementation such that it can now properly read the cpu or memory usage of a node allowing for the horizontal pod autoscaler.

This learning outcome is on proficient as I have shown that I can set up my own CI/CD environment and manage my project in such a way that it is scalable in case my application requires it. By also scanning my builds I also keep my code quality good and discover possible security risks before a branch is merged into master.

Cloud Services:

Course definition:
You integrate cloud services and serverless computing techniques that support and fit well with your enterprise application. You investigate the costs and amount of resources required for your application. Your choices for cloud provider and supported tooling are based on stakeholders' interests.

- We have built an implementation of Kubernetes with docker that should be really easy to implement within a cloud service.
- Have learned about the different design principles in a case study and what their advantages and disadvantages are in different environments. We then took these design principles and looked at what would best fit our individual project.
- Looked at the different design principles for cloud hosting and chosen what would work the best given the services I am building.
- I have my own function as a service running in the cloud. This service takes a string and changes all the curse words to stars. This service is meant to censor out all profanity in my frontend.
- With some help from Danny I managed deploy my entire project in azure cloud. Allowing for my project to be accessed from http://www.code-x.club (this is likely only up during feedback moments because I don't want to run out of azure credits too quickly)

I think this learning outcome is on proficient as I have done research on how cloud works and what kind of architectures can be deployed within the could. I have also been able to run my project in the cloud as well as been able to run a function as a service in the cloud. I think this shows that I understand and am able to work within a cloud environment.

Security by Design:

Course definition:
You integrate authentication and authorization, and mitigate possible security breaches during design and implementation of enterprise software.

- When I was working with TipTap in our group project the issue came up that it would accept plain html as content. This allowed for XSS within our project. So I had to set it up in such a way that it would not accept plain html as input. I fixed this by changing it to a by TipTap supported object like structure that would then be stored as json in the backend.
- I have an implementation of a JWT authorization service that creates and validates JWT tokens as use for user credentials when accessing user only services. For user only requests it also adds an userId header to the request before the ingress controller sends it through to the specific service allowing the controller of this userId to see the id of exactly who is trying to access certain data.
- Fixed small security issues in our group project that SonarQube was complaining about before I could merge a request.
- Now have implemented authorization within my individual project by setting a cookie containing a authorization JWT tokens.
- In semester 4 created a OWASP report about my then written application. In this report I have explained for each of the top 10 issues how it could impact my individual project and what I have done or could do to prevent these issues.

I think this learning outcome is on proficient as I have spotted and fixed security issues within my individual project and group project. I also implemented a working authorization system in Kubernetes that allows me the authorize request both on user as on roles and have made a OWASP report in which I explicitly handle each security issue of the OWASP top 10 and how it could affect my application and what I did to prevent them.

Distributed Data:

Course definition:
You are aware of data requirements and you develop enterprise systems that use distributed data tooling and best practices. You have a critical attitude towards possible privacy and ethical issues.

- We try to handle data in our group project following the guidelines of GDPR. Meaning that we won't ask for more data than necessary and are going to add the ability for an user to have their data removed if they want so.
- In our group project we are going to implement a database backup service where will make sure that if an user wants their data removed that it will also be removed from within these backups to make sure the data of our users is really gone when they ask for this.
- In my individual project I am planning on adding the ability for users the set who can see which posts and have them be able to remove any data at any point. I will try to follow the guidelines of the GDPR as much as possible.
- Have written a part of the case study about GDPR about what it entitles and what it would mean for our project.
- I have set up multiple different databases that separately contain the data for each service.
- Read the blog posts of Netflix that talk about how they handled the upscaling of their databases. By doing so I learned more about partitioning and sharding a database.
- A user can remove all of their user data if they want so. All their posts, likes and comments will be removed and no user data will be left. This is done to better follow the GDPR.
- I have learned how Cassandra works and how it could be used. I got it working in my Kubernetes cluster, however I ran into issues with within my spring boot applications that when the package was included in the POM it still could not find the requested modules and due to time constraint and two classmates also not being able to come up with a solution I decided to change to a combination of Redis and MariaDB for my trending service.
- I have implemented Redis for caching in my trending service, reducing the amount of time it takes for a request when it is cached and I am writing traffic in batches instead of immediately when there is new traffic, reducing stress on the service.
- I had a huge efficiency issue when a post was being fetched for a user. To fix this I decided to retrieve the statistics of post async. Meaning once a post is fetched in the frontend, then it will fetch that posts comment, like and revision amount reducing the time it takes for a user to see posts by almost 80%;

I think this on proficient as I have kept in mind the importance of the GDPR and have implemented features to abide to the rules of the GDPR. As well as this I have looked at different database options that can be used for different implementations and what they can be good at. On top of this I have learned about different ways of upscaling your databases if there is a need.

Semester reflection